Merck Animal Health Forecasting Project

R Code

Anqi Huang, Shuang Wei, Benjamin Elbaz and Abhishek Dixit

Table of Contents

1 - Nobivac Monthly Sales-Out Pre-Processing
2 - Nobivac Monthly Sales-In Pre-Processing
3 - Nobivac Monthly Average Inventory
4 - Nobivac Sales-Out Forecast SKU Level
5 - Nobivac Sales-In Forecast SKU Level
6 - Nobivac Sales-Out Forecast CLUSTER Level
7 - Nobivac Sales-In Forecast CLUSTER Level
8 - Nobivac Sales-Out Forecast UNIONIZED Level
9 - Nobivac Sales-In Forecast UNIONIZED Level
10 - Activyl Monthly Sales Out Pre-Processing
11 - Activyl Net Sales In and Net Sales Out from Inventory
12 - Activyl Sales-In Forecast SKU-Level

R Library Used

```r
library(zoo)
library(lattice)
library(reshape)
library(TSA)
library(forecast)
library(xts)
library(tseries)
library(lmtest)
```

1 - Nobivac Monthly Sales-Out Pre-Processing

```r
#Load the data and change the column names
data<-read.csv(file="Nobivac&Activyl_Sales_2013-2015.csv",header=TRUE)
colnames(data)[colnames(data)=="PRODUCT_CODE"]<-"Sku_Code"
colnames(data)[colnames(data)=="PRODUCT_NAME"]<-"Sku_Name"
colnames(data)[colnames(data)=="PRODUCT_FAMILY"]<-"GPF_Desc"
data$Date<-as.Date(data$INVOICE_DATE,"%d%b%Y")
data$Month<-as.Date(cut(data$Date,breaks="month"))
data$Sku_Code<-as.factor(data$Sku_Code)
n = grep("NOBIVAC",data$Sku_Name)
Nobivacdata <- data[n,]
Nob_SO_Mon_Qty<-aggregate(Nobivacdata$QUANTITY,by=list(Nobivacdata$Month,Nobivacdata$Sku_Code),sum)
names(Nob_SO_Mon_Qty)<-c("Month","Sku_Code","Quantity")


Nob_SO_Mon_Qty<-Nob_SO_Mon_Qty[which(Nob_SO_Mon_Qty$Sku_Code!="6886"
&Nob_SO_Mon_Qty$Sku_Code!="26778"&Nob_SO_Mon_Qty$Sku_Code!="36204"
&Nob_SO_Mon_Qty$Sku_Code!="37752"&Nob_SO_Mon_Qty$Sku_Code!="65299"
&Nob_SO_Mon_Qty$Sku_Code!="65299"&Nob_SO_Mon_Qty$Sku_Code!="107701"),]

convert<-read.csv(file="DosageConversion_Nobivac.csv",header=TRUE)
```

```
Nob_SO_Mon_Dose<-merge(Nob_SO_Mon_Qty,convert,by="Sku_Code")
Nob_SO_Mon_Dose$Doses<-Nob_SO_Mon_Dose$Quantity*Nob_SO_Mon_Dose$Dose
exVars <- names(Nob_SO_Mon_Dose) %in% c("Quantity","Sku_Name","Size","Dose")
Nob_SO_Mon_Dose<-Nob_SO_Mon_Dose[!exVars]

Nob_SO_Mon_Dose<-cast(Nob_SO_Mon_Dose,Month~Sku_Code,sum,value="Doses")[(1:35),]

Nob_SO_Mon.mat<-data.matrix(Nob_SO_Mon_Dose)[,(2:32)]

z.Nob_SO_Mon<-zoo(Nob_SO_Mon.mat,Nob_SO_Mon_Dose$Month)

xyplot(z.Nob_SO_Mon,type=c("b","c"),main="Monthly Sales Out Doses for Nobivac Skus",
       strip = function(bg,...)
       strip.default(bg='white',...))
```

2 - Nobivac Monthly Sales-In Pre-Processing

```
# 2013.1 - 2015.11 Sales In
Nob_SI_1<-read.csv(file="Salesin_Nobivac.csv",header=TRUE)
colnames(Nob_SI_1)[colnames(Nob_SI_1)=="UIN"]<-"Sku_Code"
colnames(Nob_SI_1)[colnames(Nob_SI_1)=="UIN_DESCRIPTION"]<-"Sku_Name"
Nob_SI_1$Date<-as.Date(Nob_SI_1$INVOICE_DATE,"%d%b%Y")
Nob_SI_1$Month<-as.Date(cut(Nob_SI_1$Date,breaks="month"))
Nob_SI_1$Sku_Code<-as.factor(Nob_SI_1$Sku_Code)
Nob_SI_1<-Nob_SI_1[which(Nob_SI_1$Month!="2015-12-01"),]

# 2015.12 - 2016.4 Sales In
New_SI<-read.csv(file="SaleIn (2015.12-2016.3).csv",header=TRUE)
n = grep("NOBIVAC",New_SI$UIN_DESCRIPTION)
Nob_SI_2<-New_SI[n,]
colnames(Nob_SI_2)[colnames(Nob_SI_2)=="UIN"]<-"Sku_Code"
colnames(Nob_SI_2)[colnames(Nob_SI_2)=="UIN_DESCRIPTION"]<-"Sku_Name"
colnames(Nob_SI_2)[colnames(Nob_SI_2)=="SALES_QUANTITY_UOM"]<-"QTY"
Nob_SI_2$Date<-as.Date(Nob_SI_2$INVOICE_DATE,"%d-%b-%y")
Nob_SI_2$Month<-as.Date(cut(Nob_SI_2$Date,breaks="month"))
Nob_SI_2$Sku_Code<-as.factor(Nob_SI_2$Sku_Code)
ExcludeVars <- names(Nob_SI_2) %in% c("DOMESTIC_SPECIES", "SALES_UOM", "DISTRIBUTOR_NUMBER")
Nob_SI_2 <- Nob_SI_2[!ExcludeVars]

Nob_SI<-rbind(Nob_SI_1,Nob_SI_2)

Nob_SI_Mon_Qty<-aggregate(Nob_SI$QTY,by=list(Nob_SI$Month,Nob_SI$Sku_Code),sum)
names(Nob_SI_Mon_Qty)<-c("Month","Sku_Code","Quantity")
Nob_SI_Mon_Qty<-Nob_SI_Mon_Qty[which(Nob_SI_Mon_Qty$Sku_Code!="6886"
&Nob_SI_Mon_Qty$Sku_Code!="26778"&Nob_SI_Mon_Qty$Sku_Code!="36204"
&Nob_SI_Mon_Qty$Sku_Code!="37752"&Nob_SI_Mon_Qty$Sku_Code!="65299"
&Nob_SI_Mon_Qty$Sku_Code!="65299"&Nob_SI_Mon_Qty$Sku_Code!="107701"),]

convert<-read.csv(file="DosageConversion_Nobivac.csv",header=TRUE)
Nob_SI_Mon_Dose<-merge(Nob_SI_Mon_Qty,convert,by="Sku_Code")
Nob_SI_Mon_Dose$Doses<-Nob_SI_Mon_Dose$Quantity*Nob_SI_Mon_Dose$Dose
exVars <- names(Nob_SI_Mon_Dose) %in% c("Quantity","Sku_Name","Size","Dose")
Nob_SI_Mon_Dose<-Nob_SI_Mon_Dose[!exVars]
Nob_SI_Mon_Dose<-cast(Nob_SI_Mon_Dose,Month~Sku_Code,sum,value="Doses")[(1:39),]

for (j in (2:32)){
  for (i in (2:38)){
    if (Nob_SI_Mon_Dose[i,j]<=0){
      Nob_SI_Mon_Dose[i,j] = 0.5*(Nob_SI_Mon_Dose[(i-1),j]+Nob_SI_Mon_Dose[(i+1),j])
    }
  }
}

Nob_SI_Mon_Dose[Nob_SI_Mon_Dose<0]<-1

Nob_SI_Mon.mat<-data.matrix(Nob_SI_Mon_Dose)[,(2:32)]

z.Nob_SI_Mon<-zoo(Nob_SI_Mon.mat,Nob_SI_Mon_Dose$Month)
xyplot(z.Nob_SI_Mon,type=c("b","c"),main="Monthly Sales In Doses for Nobivac Skus from 2013-1 to 2016-3",
       strip = function(bg,...)
         strip.default(bg='white',...))
```

3 - Nobivac Monthly Average Inventory

```
inventory = read.csv("Inventory_Dist.csv",header = TRUE)
inventory$INV_DT = as.Date(inventory$INV_DT,"%m/%d/%Y")
inventory$Month<-as.Date(cut(inventory$INV_DT,breaks="month"))
inventory$Day<-format(inventory$INV_DT,"%d")
inv_frame=data.frame(matrix(c(0:0),nrow = 35,ncol = 31))
dateSequence = seq(as.Date("2013/01/01"), by = "month", length.out = 35)
warehouse = unique(inventory$DIST_DESCRIPTION)
convert<-read.csv(file="DosageConversion_Nobivac.csv",header=TRUE)
skuCode = unique(convert$Sku_Code)
rownames(inv_frame) <- dateSequence
colnames(inv_frame) <- skuCode

Sku<-list()
for (k in (1:length(skuCode)))
{
  Sku[[k]]=inventory[grep(skuCode[k],inventory$MFGPARTNUMBER),]

  Sku_Month<-list()
  Sku_Avg_Inv = rep(0,length(dateSequence))
  for (j in (1:length(dateSequence)))
  {
    Sku_Month[[j]]=Sku[[k]][grep(dateSequence[j],Sku[[k]]$Month),]

    Sku_Dis<-list()
    uniqueWarehouse=unique(Sku_Month[[j]]$DIST_DESCRIPTION)
    temp=rep(0,length(uniqueWarehouse))
    for (i in (1:length(uniqueWarehouse)))
    {
      Sku_Dis[[i]]=Sku_Month[[j]][grep(uniqueWarehouse[i],Sku_Month[[j]]$DIST_DESCRI
                                        PTION),]
      temp[i]=(Sku_Dis[[i]]$SUM_QUANTITY_[which.min(Sku_Dis[[i]]$Day)]+Sku_Dis[[i]]$
                 SUM_QUANTITY_[which.max(Sku_Dis[[i]]$Day)])/2
    }
    Sku_Avg_Inv[j]=sum(temp)
    inv_frame[,k]=Sku_Avg_Inv
  }
}
Nob_Inv_Dose<-inv_frame
for (k in 1:length(convert$Sku_Code))
{
  Code = toString(convert$Sku_Code[k])
  Nob_Inv_Dose[,c(Code)]<-inv_frame[,c(Code)]*convert$Dose[which(convert$Sku_Code==Code)]
}
```

4 - Nobivac Sales-Out Forecast SKU Level

```r
convert<-read.csv(file="DosageConversion_Nobivac.csv",header=TRUE)
External<-read.csv(file="ExternalTotal.csv",header=TRUE)

SOResult =data.frame(matrix(c(0:0),nrow = 11,ncol = length(convert$Sku_Code)))
colnames(SOResult)=convert$Sku_Code
columnN = colnames(SOResult)

for (k in 1:length(convert$Sku_Code))
{
  Code = toString(columnN[k])
  SO_=Nob_SO_Mon_Dose[,c(Code)]
  Ext_=External$Total
  a = sum(Ext_)/sum(SO_)
  Ext_= Ext_/a

  SO_.train=ts(SO_[1:31],start=c(2013,1),frequency=12)
  Ext_.train=ts(Ext_[1:31],start=c(2013,1),frequency=12)
  G<-NULL; for(i in (1:6)){
    G[i]<-grangertest(SO_.train~Ext_.train,order=i)$`Pr(>F)`[2]
  }
  if(length(which(G<0.05))==0){
    Gind = 0
  } else{
    Gind = min(which(G<0.05))
  }

  H<-NULL; for(i in (1:6)){
    H[i]<-grangertest(Ext_.train~SO_.train,order=i)$`Pr(>F)`[2]
  }
  if(length(which(H<0.05))==0){
    Hind = 0} else{
      Hind = min(which(H<0.05))
    }

  numL = Hind
  flag = -1

  if (Gind !=0){
    numL = Gind
    flag = 1
  }

  if (flag==1){
    if(numL>=4)
    {
      Ext_.arima = auto.arima(Ext_.train,ic="aicc",trace="true")
      SO_.arima=auto.arima(SO_.train[-(1:numL)],xreg=Ext_.train[(1:(31-numL))])
      Extpredict = predict(Ext_.arima,n.ahead=(10-numL))
      SOpredict = predict(SO_.arima,newxreg=
      c(Ext_.train[((31-numL+1):length(Ext_.train))],Extpredict$pred),n.ahead=10)
    } else {
      Ext_.arima = auto.arima(Ext_.train,ic="aicc",trace="true")
      Extpredict = predict(Ext_.arima,n.ahead=(10-numL))
      SO_.arima=auto.arima(SO_.train[-(1:numL)],xreg=Ext_.train[(1:(31-numL))],trace="true")
      SOpredict = predict(SO_.arima,newxreg= c(Ext_.train[((31-numL+1):31)],Extpredict$pred),n.ahead=10)
    }
  } else {
    if(numL==0){
      SO_.arima = arima(SO_.train,c(0,1,1),seasonal=list(order=c(0,1,0),period=12))
      SOpredict = predict(SO_.arima,n.ahead = 10)
    } else{
      Ext_.arima = auto.arima(Ext_.train,ic="aicc",trace="true")
      Extpredict = predict(Ext_.arima,n.ahead=(10+numL))
      SO_.arima = auto.arima(SO_.train,xreg=c(Ext_.train[-(1:numL)],Extpredict$pred[1:numL]))
      SOpredict = predict(SO_.arima,newxreg=Extpredict$pred[((numL+1):(length(Extpredict$pred)))],n.ahead=10)
    }
  }
  len = length(SOpredict$pred)
  SOResult[,k]=as.matrix(c(SOpredict$pred[(len-9):(len)],(flag*numL)))
}
```

5 - Nobivac Sales-In Forecast SKU Level

```
convert<-read.csv(file="DosageConversion_Nobivac.csv",header=TRUE)

External<-read.csv(file="ExternalTotal.csv",header=TRUE)
SO_best = read.csv("Cluster_10Month_Sales Out Prediction.csv",check.names=FALSE)[,-1]

SIResult =data.frame(matrix(c(0:0),nrow = 10,ncol = length(convert$Sku_Code)))
colnames(SIResult)=colnames(SO_best)
columnN = colnames(SIResult)

tempResult =data.frame(matrix(c(0:0),nrow = 10,ncol = 7))
error_best = rep(NA,length(convert$Sku_Code))
model_best = rep(NA,length(convert$Sku_Code))

for (m in 1:length(convert$Sku_Code))
{
  Code = toString(columnN[m])
  SI =Nob_SI_Mon_Dose[,c(Code)]
  SO =Nob_SO_Mon_Dose[,c(Code)]
  Inv = Nob_Inv_Dose[,c(Code)]
  Ext_=External$Total
  a = sum(Ext_)/sum(SO)
  Ext_= Ext_/a
  SI.test=SI[32:39]

  SO.train=ts(SO[1:31],start=c(2013,1),frequency=12)
  Inv.train=ts(Inv[1:31],start=c(2013,1),frequency=12)
  SI.train=ts(SI[1:31],start=c(2013,1),frequency=12)
  Ext.train = ts(Ext_[1:31],start=c(2013,1),frequency=12)

  G<-NULL; for(i in (1:6)){
    G[i]<-grangertest(SI.train~Inv.train,order=i)$`Pr(>F)`[2]
  }
  if(length(which(G<0.05))==0){
    Inv_ind = 0
  } else{
    Inv_ind = min(which(G<0.05))
  }
  H<-NULL; for(i in (1:6)){
    H[i]<-grangertest(SI.train~SO.train,order=i)$`Pr(>F)`[2]
  }
  if(length(which(H<0.05))==0){
    So_ind = 0
  } else{
    So_ind = min(which(H<0.05))}
```

```r
K<-NULL; for(i in (1:6)){
  K[i]<-grangertest(SI.train~Ext.train,order=i)$`Pr(>F)`[2]
}
if(length(which(K<0.05))==0){
  Ext_ind = 0
} else{
  Ext_ind = min(which(K<0.05))}

#   So_ind
#   Inv_ind
#   Inv_ind
#   Ext_ind

error_all = rep(NA,7)
#1. ignore covariate, SI only
SI.arima = auto.arima((SI.train),ic = "aicc")
summary(SI.arima)
SIpredict = forecast(SI.arima,h=10)
SIpred = as.matrix(SIpredict$mean)

SIpred = SIpred[(length(SIpred)-9):length(SIpred)]
tempResult[,1] = SIpred

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SIpred[j]-SI.test[j])/SI.test[j]
}
error_all[1]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

#2.SI, SO, Inv
Inv.arima = auto.arima(Inv.train,xreg=fourier(Inv.train,K=6),ic = "aicc")
Invpredict = forecast.Arima(Inv.arima,xreg=fourier(Inv.train,K=6,h=10-Inv_ind))
Invpred = as.matrix(Invpredict$mean)
mine1 = (SO.train[(1+max(0,(Inv_ind-So_ind))):(length(SO.train)-So_ind)])
mine2 = (Inv.train[(1+max(0,(So_ind-Inv_ind))):(length(Inv.train)-Inv_ind)])

SI.arimax <- auto.arima((SI.train[(1+max(Inv_ind,So_ind)):(length(SI.train))]),
                        xreg=cbind(mine1,mine2),
                        ic="aicc")
SI_arimax.predict = forecast(SI.arimax,h=10,xreg=cbind(c((SO.train[-(1:(length(SO.train)-So_ind))]),
                                                         (SO_best[,c(Code)][1:(10-So_ind)])),
                                                       c((Inv.train[-(1:(length(Inv.train)-Inv_ind))]),
                                                         (Invpred))))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

SI_arimax.pred = SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]
tempResult[,2] = SI_arimax.pred

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error_all[2]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2


#3.SI, SO
SI.arimax <- auto.arima((SI.train[(1+So_ind):(length(SI.train))])
                        ,xreg=(SO.train[1:(length(SO.train)-So_ind)]),ic="aicc")
SI_arimax.predict = forecast(SI.arimax,h=10,xreg=c((SO.train[-(1:(length(SO.train)-So_ind))])
                        ,(SO_best[,c(Code)][1:(10-So_ind)])))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

SI_arimax.pred =SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]
tempResult[,3] = SI_arimax.pred

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error_all[3]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2


#4.SI, Inv
Inv.arima = auto.arima(Inv.train,xreg=fourier(Inv.train,K=6),ic = "aicc")
Invpredict = forecast(Inv.arima,xreg=fourier(Inv.train,K=6,h=(10-Inv_ind)))
Invpred = as.matrix(Invpredict$mean)

SI.arimax <- auto.arima((SI.train[(1+Inv_ind):length(SI.train)])
                        ,xreg=(Inv.train[1:(length(Inv.train)-Inv_ind)]),ic="aicc",trace=TRUE)
SI_arimax.predict = forecast(SI.arimax,h=10
                        ,xreg=c((Inv.train[-(1:(length(Inv.train)-Inv_ind))]),(Invpred)))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

SI_arimax.pred = SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]
tempResult[,4] = SI_arimax.pred

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error_all[4]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2



#5. SI, Ext, Inv
Inv.arima = auto.arima(Inv.train,xreg=fourier(Inv.train,K=6),ic = "aicc")
Invpredict = forecast(Inv.arima,xreg=fourier(Inv.train,K=6,h=10-Inv_ind))
Invpred = as.matrix(Invpredict$mean)

Ext_.arima = auto.arima(Ext.train,ic="aicc",trace="true")
Extpredict = predict(Ext_.arima,n.ahead=(10-Ext_ind))
new1 = (Inv.train[1:(length(Inv.train)-Inv_ind)][(1+max(0,(Ext_ind-Inv_ind))):(length(Inv.train)-Inv_ind)])
new2 = (Ext.train[1:(length(Ext.train)-Ext_ind)][(1+max(0,(Inv_ind-Ext_ind))):(length(Ext.train)-Ext_ind)])

SI.arimax <- auto.arima((SI.train[(1+max(Ext_ind,Inv_ind)):(length(SI.train))]),
                        xreg=cbind(new1,new2),
                        ic="aicc",trace=TRUE)
SI_arimax.predict = forecast(SI.arimax,h=10,xreg=cbind(c((Inv.train[-(1:(length(Inv.train)-Inv_ind))]),
                                                          (Invpred)),
                                                        c((Ext.train[-(1:(length(Ext.train)-Ext_ind))]),
                                                          (Extpredict$pred))))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

SI_arimax.pred=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]
tempResult[,5] = SI_arimax.pred

err = rep(NA,8)
for (j in (1:8))
```

```
  {
    err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
  }
  error_all[5]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2


  #6. SI, Ext
  Ext_.arima = auto.arima(Ext.train,ic="aicc",trace="true")
  Extpredict = predict(Ext_.arima,n.ahead=(10-Ext_ind))

  SI.arimax <- auto.arima((SI.train[(1+Ext_ind):(length(SI.train))])
                         ,xreg=(Ext.train[1:(length(Ext.train)-Ext_ind)]),ic="aicc",trace=TRUE)
  SI_arimax.predict = forecast(SI.arimax,h=10
                         ,xreg=c((Ext.train[-(1:(length(Ext.train)-Ext_ind))]),(Extpredict$pred)))
  SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

  SI_arimax.pred = SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]
  tempResult[,6] = SI_arimax.pred

  err = rep(NA,8)
  for (j in (1:8))
  {
    err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
  }
  error_all[6]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2


  #7. SI, Ext, SO
  Ext_.arima = auto.arima(Ext.train,ic="aicc",trace="true")
  Extpredict = predict(Ext_.arima,n.ahead=(10-Ext_ind))
  cov1 = (SO.train[(1+max(0,(Ext_ind-So_ind))):(length(SO.train)-So_ind)])
  cov2 = (Ext.train[(1+max(0,(So_ind-Ext_ind))):(length(Ext.train)-Ext_ind)])

  SI.arimax <- auto.arima((SI.train[(1+max(Ext_ind,So_ind)):(length(SI.train))]),
                         xreg=cbind(cov1,cov2),
                         ic="aicc",trace=TRUE)
  SI_arimax.predict = forecast(SI.arimax,h=10,xreg=cbind(c((SO.train[-(1:(length(SO.train)-So_ind))]),
                                                          (SO_best[,c(Code)][1:(10-So_ind)])),
                                                        c((Ext.train[-(1:(length(Ext.train)-Ext_ind))]),
                                                          (Extpredict$pred))))
  SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

  SI_arimax.pred = SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]
  tempResult[,7] = SI_arimax.pred

  err = rep(NA,8)
  for (j in (1:8))
  {
    err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
  }
  error_all[7]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2


  modelind<-1:7
  error_best[m]=min(error_all)
  model_best[m]=modelind[which(error_all==min(error_all))]

  SIResult[,m]=tempResult[,which(error_all==min(error_all))]
}
SIResult<-rbind(SIResult,error_best,model_best)
```

6 - Nobivac Sales-Out Forecast CLUSTER Level

```r
convert<-read.csv(file="DosageConversion_Nobivac.csv",header=TRUE)
External<-read.csv(file="ExternalTotal.csv",header=TRUE)

columnN = c('Cluster1','Cluster2','Cluster3','Cluster4','Cluster5')

##### Cluster 3 #####
k=1  # change k
Code = toString(columnN[k])
SO_=Nob_SO_Mon_Dose[,c(Code)]
Ext_=External$Total
a = sum(Ext_)/sum(SO_)
Ext_= Ext_/a

SO_.train=ts(SO_[1:31],start=c(2013,1),frequency=12)
Ext_.train=ts(Ext_[1:31],start=c(2013,1),frequency=12)
G<-NULL; for(i in (1:6)){
  G[i]<-grangertest(SO_.train~Ext_.train,order=i)$`Pr(>F)`[2]
}
if(length(which(G<0.05))==0){
  Gind = 0
} else{
  Gind = min(which(G<0.05))
}

H<-NULL; for(i in (1:6)){
  H[i]<-grangertest(Ext_.train~SO_.train,order=i)$`Pr(>F)`[2]
}
if(length(which(H<0.05))==0){
  Hind = 0} else{
    Hind = min(which(H<0.05))
  }

numL = Hind
flag = -1

if (Gind !=0){
  numL = Gind
  flag = 1
}

if (flag==1){
  if(numL>=4)
  {
    Ext_.arima = auto.arima(Ext_.train,ic="aicc",trace="true")
    SO_.arima=auto.arima(SO_.train[-(1:numL)],xreg=Ext_.train[(1:(31-numL))])
    Extpredict = predict(Ext_.arima,n.ahead=(10-numL))
    SOpredict = predict(SO_.arima,newxreg= c(Ext_.train[((31-numL+1):length(Ext_.train))],
                                            Extpredict$pred),n.ahead=10)
  } else {
    Ext_.arima = auto.arima(Ext_.train,ic="aicc",trace="true")
    Extpredict = predict(Ext_.arima,n.ahead=(10-numL))
    SO_.arima=auto.arima(SO_.train[-(1:numL)],xreg=Ext_.train[(1:(31-numL))],trace="true")
    SOpredict = predict(SO_.arima,newxreg= c(Ext_.train[((31-numL+1):31)],Extpredict$pred)
                        ,n.ahead=10)
  }
} else {
  if(numL==0){
    SO_.arima = arima(SO_.train,c(0,1,1),seasonal=list(order=c(0,1,0),period=12))
    SOpredict = predict(SO_.arima,n.ahead = 10)
  } else{
    Ext_.arima = auto.arima(Ext_.train,ic="aicc",trace="true")
    Extpredict = predict(Ext_.arima,n.ahead=(10+numL))
    SO_.arima = auto.arima(SO_.train,xreg=c(Ext_.train[-(1:numL)],Extpredict$pred[1:numL]))
    SOpredict = predict(SO_.arima,
                        newxreg=Extpredict$pred[((numL+1):(length(Extpredict$pred)))],n.ahead=10)
  }
}

flag*numL
summary(SO_.arima)

# Cluster 1 Sales Out Result
Result = data.frame(matrix(c(0:0),nrow = 41,ncol = (length(colnames(ClusterSO1))-1))) # change Cluster
colnames(Result)<-colnames(ClusterSO1)[-length(colnames(ClusterSO1))] # change Cluster
for(i in (1:(length(colnames(ClusterSO1))-1))){ # change Cluster
  pred = c(ClusterSO1[(1:31),i],rep(0,10)) # change Cluster
  b = sum(ClusterSO1$sum)/sum(ClusterSO1[,i]) # change Cluster
  Ext_.pred = c(Ext_.train,Extpredict$pred)/b
  for(j in (32:41)){
    pred[j] = pred[j-1]-0.4925*(Ext_.pred[j-2]-Ext_.pred[j-3])
  }
  Result[,i]=as.matrix(pred)
}

# Cluster 2 Sales Out Result
Result = data.frame(matrix(c(0:0),nrow = 41,ncol = (length(colnames(ClusterSO2))-1)))
colnames(Result)<-colnames(ClusterSO2)[-length(colnames(ClusterSO2))]
for(i in (1:(length(colnames(ClusterSO2))-1))){
  pred = c(ClusterSO2[(1:31),i],rep(0,10))
  b = sum(ClusterSO2$sum)/sum(ClusterSO2[,i])
  Ext_.pred = c(Ext_.train,Extpredict$pred)/b
  for(j in (32:41)){
    pred[j] = (1/b)*78943.26+0.4730*Ext_.pred[j-3]
  }
  Result[,i]=as.matrix(pred)
}

# Cluster 3 Sales Out Result
w = SO_.arima$residuals
w = c(rep(0,numL),w,rep(0,10))
Result = data.frame(matrix(c(0:0),nrow = 41,ncol = (length(colnames(ClusterSO3))-1)))
colnames(Result)<-colnames(ClusterSO3)[-length(colnames(ClusterSO3))]
for(i in (1:(length(colnames(ClusterSO3))-1))){
  pred = c(ClusterSO3[(1:31),i],rep(0,10))
  b = sum(ClusterSO3$sum)/sum(ClusterSO3[,i])
  Ext_.pred = c(Ext_.train,Extpredict$pred)/b
  for(j in (32:41)){
    pred[j] = pred[j-1]+0.4755*(Ext_.pred[j-3]-Ext_.pred[j-4])-0.7729*w[j-1]
  }
  Result[,i]=as.matrix(pred)
}

# Cluster 4 Sales Out Result
Result = data.frame(matrix(c(0:0),nrow = 41,ncol = (length(colnames(ClusterSO4))-1)))
colnames(Result)<-colnames(ClusterSO4)[-length(colnames(ClusterSO4))]
for(i in (1:(length(colnames(ClusterSO4))-1))){
  pred = c(ClusterSO4[(1:31),i],rep(0,10))
  b = sum(ClusterSO4$sum)/sum(ClusterSO4[,i])
  Ext_.pred = c(Ext_.train,Extpredict$pred)/b
```

```r
  for(j in (32:41)){
    pred[j] = (1/b)*1298273.2+0.2185*Ext_.pred[j-1]
  }
  Result[,i]=as.matrix(pred)
}


# Cluster 5 Sales Out Result
Result = data.frame(matrix(c(0:0),nrow = 41,ncol = (length(colnames(ClusterSO5))-1)))
colnames(Result)<-colnames(ClusterSO5)[-length(colnames(ClusterSO5))]
for(i in (1:(length(colnames(ClusterSO5))-1))){
  pred = c(ClusterSO5[(1:31),i],rep(0,10))
  b = sum(ClusterSO5$sum)/sum(ClusterSO5[,i])
  Ext_.pred = c(Ext_.train,Extpredict$pred)/b
  for(j in (32:41)){
    pred[j] = (1-0.8049)*pred[j-1]+(0.8049-0.6890)*pred[j-2]
              +0.6890*pred[j-3]-0.5243*(Ext_.pred[j-5]-Ext_.pred[j-6])
  }
  Result[,i]=as.matrix(pred)
}
```

7 - Nobivac Sales-In Forecast CLUSTER Level

```r
convert<-read.csv(file="DosageConversion_Nobivac.csv",header=TRUE)

External<-read.csv(file="ExternalTotal.csv",header=TRUE)

SO_best = read.csv("Cluster_10Month_Sales Out Prediction.csv",check.names=FALSE)[,-1]
SO_best$Cluster1<-rowSums(SO_best[c('54215','99895','65448','65447')])
SO_best$Cluster2<-rowSums(SO_best[c('54219','65266','31664')])
SO_best$Cluster3<-rowSums(SO_best[c('65264','65265','65277','65267')])
SO_best$Cluster4<-rowSums(SO_best[c('65288','65289','65310','65313',
                                    '65314','65316','65440','65290','65293','65300',
                                    '65441','65315','6772','65284','65444')])
SO_best$Cluster5<-rowSums(SO_best[c('68616','99336')])

SIResult=data.frame(matrix(c(0:0),nrow = 10,ncol = 7))
columnN =c('Cluster1','Cluster2','Cluster3','Cluster4','Cluster5')

error = matrix(0,nrow=5,ncol=7,byrow=TRUE)

##### 7 SI Models on 5 Clusters
for (k in (1:5)){
  k=1
  Code = toString(columnN[k])
  SI =Nob_SI_Mon_Dose[,c(Code)]
  SO =Nob_SO_Mon_Dose[,c(Code)]
  Inv = Nob_Inv_Dose[,c(Code)]
  Ext =External$Total
  a = sum(Ext)/sum(SI)
  Ext = Ext/a
  SI.test=SI[32:39]

  SO.train=ts(SO[1:31],start=c(2013,1),frequency=12)
  Inv.train=ts(Inv[1:31],start=c(2013,1),frequency=12)
  SI.train=ts(SI[1:31],start=c(2013,1),frequency=12)
  Ext.train = ts(Ext[1:31],start=c(2013,1),frequency=12)

  G<-NULL; for(i in (1:6)){
```

```
  G[i]<-grangertest(SI.train~Inv.train,order=i)$`Pr(>F)`[2]
}
if(length(which(G<0.05))==0){
  Inv_ind = 0
} else{
  Inv_ind = min(which(G<0.05))
}
H<-NULL; for(i in (1:6)){
  H[i]<-grangertest(SI.train~SO.train,order=i)$`Pr(>F)`[2]
}
if(length(which(H<0.05))==0){
  So_ind = 0
} else{
  So_ind = min(which(H<0.05))}
K<-NULL; for(i in (1:6)){
  K[i]<-grangertest(SI.train~Ext.train,order=i)$`Pr(>F)`[2]
}
if(length(which(K<0.05))==0){
  Ext_ind = 0
} else{
  Ext_ind = min(which(K<0.05))}

So_ind
Inv_ind
Ext_ind

#1. ignore covariate, SI only
SI.arima = auto.arima((SI.train),ic = "aicc")
summary(SI.arima)
SIpredict = forecast(SI.arima,h=10)
SIpred = as.matrix(SIpredict$mean)

SIResult[,1]=SIpred[(length(SIpred)-9):length(SIpred)]
err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SIpred[j]-SI.test[j])/SI.test[j]
}
error[k,1]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

#2.SI, SO, Inv

Inv.arima = auto.arima(Inv.train,xreg=fourier(Inv.train,K=6),ic = "aicc")
Invpredict = forecast(Inv.arima,xreg=fourier(Inv.train,K=6,h=10-Inv_ind))
Invpred = as.matrix(Invpredict$mean)
mine1 = (SO.train[(1+max(0,(Inv_ind-So_ind))):(length(SO.train)-So_ind)])
mine2 = (Inv.train[(1+max(0,(So_ind-Inv_ind))):(length(Inv.train)-Inv_ind)])

SI.arimax <- auto.arima((SI.train[(1+max(Inv_ind,So_ind)):(length(SI.train))]),
                        xreg=cbind(mine1,mine2),
                        ic="aicc")
summary(SI.arimax)
SI_arimax.predict = forecast(SI.arimax,h=10,xreg=cbind(c((SO.train[-(1:(length(SO.train)-So_ind))]),
                              (SO_best[,c(Code)][1:(10-So_ind)])),
                              c((Inv.train[-(1:(length(Inv.train)-Inv_ind))]),
                               (Invpred))))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)
SI_arimax.pred = (SI_arimax.pred)

SIResult[,2]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]
```

```
err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error[k,2]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

#3.SI, SO
SI.arimax <- auto.arima((SI.train[(1+So_ind):(length(SI.train))]),
                        xreg=(SO.train[1:(length(SO.train)-So_ind)]),ic="aicc")
SI_arimax.predict = forecast(SI.arimax,h=10,
                        xreg=c((SO.train[-(1:(length(SO.train)-So_ind))]),
                        (SO_best[,c(Code)][1:(10-So_ind)])))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)
SI_arimax.pred = (SI_arimax.pred)

SIResult[,3]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error[k,3]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

#4.SI, Inv
Inv.arima = auto.arima(Inv.train,xreg=fourier(Inv.train,K=6),ic = "aicc")
Invpredict = forecast(Inv.arima,xreg=fourier(Inv.train,K=6,h=(10-Inv_ind)))
Invpred = as.matrix(Invpredict$mean)

SI.arimax <- auto.arima((SI.train[(1+Inv_ind):length(SI.train)]),
                        xreg=(Inv.train[1:(length(Inv.train)-Inv_ind)]),
                        ic="aicc",trace=TRUE)
SI_arimax.predict = forecast(SI.arimax,h=10,
                        xreg=c((Inv.train[-(1:(length(Inv.train)-Inv_ind))]),
                        (Invpred)))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)
SI_arimax.pred = (SI_arimax.pred)

SIResult[,4]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error[k,4]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

#5. SI, Ext, Inv
Inv.arima = auto.arima(Inv.train,xreg=fourier(Inv.train,K=6),ic = "aicc")
Invpredict = forecast(Inv.arima,xreg=fourier(Inv.train,K=6,h=10-Inv_ind))
Invpred = as.matrix(Invpredict$mean)

Ext_.arima = auto.arima(Ext.train,ic="aicc",trace="true")
Extpredict = predict(Ext_.arima,n.ahead=(10-Ext_ind))
new1 = (Inv.train[1:(length(Inv.train)-Inv_ind)][(1+max(0,(Ext_ind-Inv_ind))):(length(Inv.train)-Inv_ind)])
new2 = (Ext.train[1:(length(Ext.train)-Ext_ind)][(1+max(0,(Inv_ind-Ext_ind))):(length(Ext.train)-Ext_ind)])

SI.arimax <- auto.arima((SI.train[(1+max(Ext_ind,Inv_ind)):(length(SI.train))]),
                        xreg=cbind(new1,new2),
                        ic="aicc",trace=TRUE)
```

```r
SI_arimax.predict = forecast(SI.arimax,h=10,
                             xreg=cbind(c((Inv.train[-(1:(length(Inv.train)-Inv_ind))]),
                             (Invpred)),
                             c((Ext.train[-(1:(length(Ext.train)-Ext_ind))]),
                             (Extpredict$pred))))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)
SI_arimax.pred = (SI_arimax.pred)

summary(SI.arimax)
SIResult[,5]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]

err = rep(NA,8)

for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error[k,5]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

#6. SI, Ext
Ext_.arima = auto.arima(Ext.train,ic="aicc",trace="true")
Extpredict = predict(Ext_.arima,n.ahead=(10-Ext_ind))

SI.arimax <- auto.arima((SI.train[(1+Ext_ind):(length(SI.train))]),
                        xreg=(Ext.train[1:(length(Ext.train)-Ext_ind)]),
                        ic="aicc",trace=TRUE)
SI_arimax.predict = forecast(SI.arimax,h=10,
                        xreg=c((Ext.train[-(1:(length(Ext.train)-Ext_ind))]),
                        (Extpredict$pred)))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)
SI_arimax.pred = (SI_arimax.pred)

SIResult[,6]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error[k,6]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2


#7. SI, Ext, SO
Ext_.arima = auto.arima(Ext.train,ic="aicc",trace="true")
Extpredict = predict(Ext_.arima,n.ahead=(10-Ext_ind))
cov1 = (SO.train[(1+max(0,(Ext_ind-So_ind))):(length(SO.train)-So_ind)])
cov2 = (Ext.train[(1+max(0,(So_ind-Ext_ind))):(length(Ext.train)-Ext_ind)])

SI.arimax <- auto.arima((SI.train[(1+max(Ext_ind,So_ind)):(length(SI.train))]),
                        xreg=cbind(cov1,cov2),
                        ic="aicc",trace=TRUE)
SI_arimax.predict = forecast(SI.arimax,h=10,
                             xreg=cbind(c((SO.train[-(1:(length(SO.train)-So_ind))]),
                             (SO_best[,c(Code)][1:(10-So_ind)])),
                             c((Ext.train[-(1:(length(Ext.train)-Ext_ind))]),
                             (Extpredict$pred))))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)
SI_arimax.pred = (SI_arimax.pred)

SIResult[,7]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]
```

```
  err = rep(NA,8)
  for (j in (1:8))
  {
    err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
  }
  error[k,7]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2
}

colnames(error)<-c("SI Only",
                   "SI, SO, Inv",
                   "SI, SO",
                   "SI, Inv",
                   "SI, Ext, Inv",
                   "SI, Ext",
                   "SI, Ext, SO")
rownames(error)<-c("Cluster 1", "Cluster 2", "Cluster 3", "Cluster 4", "Cluster 5")

# Cluster 1 (SI,Ext,SO)
Result = data.frame(matrix(c(0:0),nrow = 41,ncol = (length(colnames(ClusterSI1))-1)))
colnames(Result)<-colnames(ClusterSI1)[-length(colnames(ClusterSI1))]
for(i in (1:(length(colnames(ClusterSI1))-1))){
  pred = c(ClusterSI1[(1:31),i],rep(0,10))
  b = sum(ClusterSI1$sum)/sum(ClusterSI1[,i])
  Ext_.pred = c(Ext.train,Extpredict$pred)/b
  SO_.pred = SO_best$Cluster1/b
  for(j in (32:41)){
    pred[j] = (1-0.8289)*pred[j-1]+(-0.6178+0.8289)*pred[j-2]
    +0.6178*pred[j-3]+0.3912*(SO_.pred[j-1]-SO_.pred[j-2])
    +0.4598*(Ext_.pred[j]-Ext_.pred[j-1])
  }
  Result[,i]=as.matrix(pred)
}

# Cluster 2 (SI,Ext,SO)
Result = data.frame(matrix(c(0:0),nrow = 41,ncol = (length(colnames(ClusterSI2))-1)))
colnames(Result)<-colnames(ClusterSI2)[-length(colnames(ClusterSI2))]
for(i in (1:(length(colnames(ClusterSI2))-1))){
  pred = c(ClusterSI2[(1:31),i],rep(0,10))
  b = sum(ClusterSI2$sum)/sum(ClusterSI2[,i])
  Ext_.pred = c(Ext.train,Extpredict$pred)/b
  SO_.pred = SO_best$Cluster2/b
  for(j in (32:41)){
    pred[j] = -41610.74/b+0.9238*SO_.pred[j-1]+0.3624*Ext_.pred[j-3]
  }
  Result[,i]=as.matrix(pred)
}


# Cluster 3 (SI,Ext)
Result = data.frame(matrix(c(0:0),nrow = 41,ncol = (length(colnames(ClusterSI3))-1)))
colnames(Result)<-colnames(ClusterSI3)[-length(colnames(ClusterSI3))]
for(i in (1:(length(colnames(ClusterSI3))-1))){
  pred = c(ClusterSI3[(1:31),i],rep(0,10))
  b = sum(ClusterSI3$sum)/sum(ClusterSI3[,i])
  Inv_.pred = c(Inv.train,Invpredict$mean)/b
  for(j in (32:41)){
    pred[j] = 230926.13/b+0.2421*Inv_.pred[j-5]
  }
  Result[,i]=as.matrix(pred)
}
```

```r
# Cluster 4 (SI,SO,Inv)
Result = data.frame(matrix(c(0:0),nrow = 41,ncol = (length(colnames(ClusterSI4))-1)))
colnames(Result)<-colnames(ClusterSI4)[-length(colnames(ClusterSI4))]
for(i in (1:(length(colnames(ClusterSI4))-1))){
  pred = c(ClusterSI4[(1:31),i],rep(0,10))
  b = sum(ClusterSI4$sum)/sum(ClusterSI4[,i])
  SO_.pred = SO_best$Cluster4/b
  Inv_.pred = c(Inv.train,Invpredict$mean)/b
  for(j in (32:41)){
    pred[j] = 3.7783*SO_.pred[j]-0.3985*Inv_.pred[j]
  }
  Result[,i]=as.matrix(pred)
}

# Cluster 5 (SI)
Result = data.frame(matrix(c(0:0),nrow = 41,ncol = (length(colnames(ClusterSI5))-1)))
colnames(Result)<-colnames(ClusterSI5)[-length(colnames(ClusterSI5))]
for(i in (1:(length(colnames(ClusterSI5))-1))){
  b = sum(ClusterSI5$sum)/sum(ClusterSI5[,i])
  pred = c(ClusterSI5[(1:31),i],SIpred/b)
  Result[,i]=as.matrix(pred)
}
```

8 - Nobivac Sales-Out Forecast UNIONIZED Level

```r
convert<-read.csv(file="DosageConversion_Nobivac.csv",header=TRUE)
External<-read.csv(file="ExternalTotal.csv",header=TRUE)

SO_=Nob_SO_Mon_Dose$Aggregate
Ext_=External$Total
a = sum(Ext_)/sum(SO_)
Ext_= Ext_/a

SO_.train=ts(SO_[1:31],start=c(2013,1),frequency=12)
Ext_.train=ts(Ext_[1:31],start=c(2013,1),frequency=12)
G<-NULL; for(i in (1:6)){
  G[i]<-grangertest(SO_.train~Ext_.train,order=i)$`Pr(>F)`[2]
}
if(length(which(G<0.05))==0){
  Gind = 0
} else{
  Gind = min(which(G<0.05))
}

H<-NULL; for(i in (1:6)){
  H[i]<-grangertest(Ext_.train~SO_.train,order=i)$`Pr(>F)`[2]
}
if(length(which(H<0.05))==0){
  Hind = 0} else{
    Hind = min(which(H<0.05))
  }

numL = Hind
flag = -1

if (Gind !=0){
  numL = Gind
  flag = 1
}
```

```r
if (flag==1){
  if(numL>=4)
  {
    Ext_.arima = auto.arima(Ext_.train,ic="aicc",trace="true")
    SO_.arima=auto.arima(SO_.train[-(1:numL)],xreg=Ext_.train[(1:(31-numL))])
    Extpredict = predict(Ext_.arima,n.ahead=(10-numL))
    SOpredict = predict(SO_.arima,
                newxreg=c(Ext_.train[((31-numL+1):length(Ext_.train))],
                Extpredict$pred),n.ahead=10)
  } else {
    Ext_.arima = auto.arima(Ext_.train,ic="aicc",trace="true")
    Extpredict = predict(Ext_.arima,n.ahead=(10-numL))
    SO_.arima=auto.arima(SO_.train[-(1:numL)],xreg=Ext_.train[(1:(31-numL))],trace="true")
    SOpredict = predict(SO_.arima,newxreg= c(Ext_.train[((31-numL+1):31)],Extpredict$pred),n.ahead=10)
  }
} else {
  if(numL==0){
    SO_.arima = arima(SO_.train,c(0,1,1),seasonal=list(order=c(0,1,0),period=12))
    SOpredict = predict(SO_.arima,n.ahead = 10)
  } else{
    Ext_.arima = auto.arima(Ext_.train,ic="aicc",trace="true")
    Extpredict = predict(Ext_.arima,n.ahead=(10+numL))
    SO_.arima = auto.arima(SO_.train,xreg=c(Ext_.train[-(1:numL)],Extpredict$pred[1:numL]))
    SOpredict = predict(SO_.arima,
                newxreg=Extpredict$pred[((numL+1):(length(Extpredict$pred)))],
                n.ahead=10)
  }
}

flag*numL
summary(SO_.arima)

Result = data.frame(matrix(c(0:0),nrow = 41,ncol = 31))
colnames(Result)<-colnames(Nob_SI_Mon_Dose)[2:32]
for(i in (1:31)){
  pred = c(Nob_SO_Mon_Dose[(1:31),(i+1)],rep(0,10))
  b = sum(Nob_SO_Mon_Dose$Aggregate)/sum(Nob_SO_Mon_Dose[,(i+1)])
  Ext_.pred = c(Ext_.train,Extpredict$pred)/b
  for(j in (32:41)){
    pred[j] = (1-0.8056)*pred[j-1]+(0.8056-0.6347)*pred[j-2]
    +0.6347*pred[j-3]+0.2644*(Ext_.pred[j-1]-Ext_.pred[j-2])
  }
  Result[,i]=as.matrix(pred)
}
```

9 - Nobivac Sales-In Forecast UNIONIZED Level

```r
convert<-read.csv(file="DosageConversion_Nobivac.csv",header=TRUE)

External<-read.csv(file="ExternalTotal.csv",header=TRUE)

SO_best = read.csv("Cluster_10Month_Sales Out Prediction.csv",check.names=FALSE)[,-1]
SO_best$Aggregate<-rowSums(SO_best[,(1:31)])

SIResult=data.frame(matrix(c(0:0),nrow = 10,ncol = 7))

error = rep(NA,7)
```

```r
##### 7 SI Models on 5 Clusters
  Code = "Aggregate"
  SI =Nob_SI_Mon_Dose[,c(Code)]
  SO =Nob_SO_Mon_Dose[,c(Code)]
  Inv = Nob_Inv_Dose[,c(Code)]
  Ext_=External$Total
  SI.test=SI[32:39]

  SO.train=ts(SO[1:35],start=c(2013,1),frequency=12)
  Inv.train=ts(Inv[1:35],start=c(2013,1),frequency=12)
  SI.train=ts(SI[1:35],start=c(2013,1),frequency=12)
  Ext.train = ts(Ext_[1:35],start=c(2013,1),frequency=12)

  G<-NULL; for(i in (1:6)){
    G[i]<-grangertest(SI.train~Inv.train,order=i)$`Pr(>F)`[2]
  }
  if(length(which(G<0.05))==0){
    Inv_ind = 0
  } else{
    Inv_ind = min(which(G<0.05))
  }
  H<-NULL; for(i in (1:6)){
    H[i]<-grangertest(SI.train~SO.train,order=i)$`Pr(>F)`[2]
  }
  if(length(which(H<0.05))==0){
    So_ind = 0
  } else{
    So_ind = min(which(H<0.05))}
  K<-NULL; for(i in (1:6)){
    K[i]<-grangertest(SI.train~Ext.train,order=i)$`Pr(>F)`[2]
  }
  if(length(which(K<0.05))==0){
    Ext_ind = 0
  } else{
    Ext_ind = min(which(K<0.05))}

  So_ind
  Inv_ind
  Ext_ind

  #1. ignore covariate, SI only
  SI.arima = auto.arima((SI.train),ic = "aicc")
  summary(SI.arima)
  SIpredict = forecast(SI.arima,h=10)
  SIpred = as.matrix(SIpredict$mean)

  SIResult[,1]=SIpred[(length(SIpred)-9):length(SIpred)]
  err = rep(NA,8)
  for (j in (1:8))
  {
    err[j]=abs(SIpred[j]-SI.test[j])/SI.test[j]
  }
  error[1]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

  #2.SI, SO, Inv

  Inv.arima = auto.arima(Inv.train,xreg=fourier(Inv.train,K=6),ic = "aicc")
  Invpredict = forecast(Inv.arima,xreg=fourier(Inv.train,K=6,h=10-Inv_ind))
  Invpred = as.matrix(Invpredict$mean)
  mine1 = (SO.train[(1+max(0,(Inv_ind-So_ind))):(length(SO.train)-So_ind)])
  mine2 = (Inv.train[(1+max(0,(So_ind-Inv_ind))):(length(Inv.train)-Inv_ind)])
```

```r
SI.arimax <- auto.arima((SI.train[(1+max(Inv_ind,So_ind)):(length(SI.train))]),
                        xreg=cbind(mine1,mine2),
                        ic="aicc")
SI_arimax.predict = forecast(SI.arimax,h=10,xreg=cbind(c((SO.train[-(1:(length(SO.train)-So_ind))]),
                          (SO_best[,c(Code)][1:(10-So_ind)])),
                          c((Inv.train[-(1:(length(Inv.train)-Inv_ind))]),
                          (Invpred))))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

SIResult[,2]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error[2]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

#3.SI, SO
SI.arimax <- auto.arima((SI.train[(1+So_ind):(length(SI.train))]),
                        xreg=(SO.train[1:(length(SO.train)-So_ind)]),
                        ic="aicc")
SI_arimax.predict = forecast(SI.arimax,h=10,
                        xreg=c((SO.train[-(1:(length(SO.train)-So_ind))]),
                        (SO_best[,c(Code)][1:(10-So_ind)])))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

SIResult[,3]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error[3]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

#4.SI, Inv
Inv.arima = auto.arima(Inv.train,xreg=fourier(Inv.train,K=6),ic = "aicc")
Invpredict = forecast(Inv.arima,xreg=fourier(Inv.train,K=6,h=(10-Inv_ind)))
Invpred = as.matrix(Invpredict$mean)

SI.arimax <- auto.arima((SI.train[(1+Inv_ind):length(SI.train)]),
                  xreg=(Inv.train[1:(length(Inv.train)-Inv_ind)]),
                  ic="aicc",trace=TRUE)
SI_arimax.predict = forecast(SI.arimax,h=10,
                  xreg=c((Inv.train[-(1:(length(Inv.train)-Inv_ind))]),
                  (Invpred)))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

SIResult[,4]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error[4]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

#5. SI, Ext, Inv
Inv.arima = auto.arima(Inv.train,xreg=fourier(Inv.train,K=6),ic = "aicc")
```

```
Invpredict = forecast(Inv.arima,xreg=fourier(Inv.train,K=6,h=10-Inv_ind))
Invpred = as.matrix(Invpredict$mean)

Ext_.arima = auto.arima(Ext.train,ic="aicc",trace="true")
Extpredict = predict(Ext_.arima,n.ahead=(10-Ext_ind))
new1 = (Inv.train[1:(length(Inv.train)-Inv_ind)][(1+max(0,(Ext_ind-Inv_ind))):(length(Inv.train)-Inv_ind)])
new2 = (Ext.train[1:(length(Ext.train)-Ext_ind)][(1+max(0,(Inv_ind-Ext_ind))):(length(Ext.train)-Ext_ind)])

SI.arimax <- auto.arima((SI.train[(1+max(Ext_ind,Inv_ind)):(length(SI.train))]),
                        xreg=cbind(new1,new2),
                        ic="aicc",trace=TRUE)
SI_arimax.predict = forecast(SI.arimax,h=10,xreg=cbind(c((Inv.train[-(1:(length(Inv.train)-Inv_ind))]),
                                                          (Invpred)),
                                                        c((Ext.train[-(1:(length(Ext.train)-Ext_ind))]),
                                                          (Extpredict$pred))))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

SIResult[,5]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error[5]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

#6. SI, Ext
Ext_.arima = auto.arima(Ext.train,ic="aicc",trace="true")
Extpredict = predict(Ext_.arima,n.ahead=(10-Ext_ind))

SI.arimax <- auto.arima((SI.train[(1+Ext_ind):(length(SI.train))]),
                      xreg=(Ext.train[1:(length(Ext.train)-Ext_ind)]),
                      ic="aicc",trace=TRUE)
SI_arimax.predict = forecast(SI.arimax,h=10,
                      xreg=c((Ext.train[-(1:(length(Ext.train)-Ext_ind))]),
                      (Extpredict$pred)))
SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

SIResult[,6]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]

err = rep(NA,8)
for (j in (1:8))
{
  err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
}
error[6]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2


#7. SI, Ext, SO
Ext_.arima = auto.arima(Ext.train,ic="aicc",trace="true")
Extpredict = predict(Ext_.arima,n.ahead=(10-Ext_ind))
cov1 = (SO.train[(1+max(0,(Ext_ind-So_ind))):(length(SO.train)-So_ind)])
cov2 = (Ext.train[(1+max(0,(So_ind-Ext_ind))):(length(Ext.train)-Ext_ind)])

SI.arimax <- auto.arima((SI.train[(1+max(Ext_ind,So_ind)):(length(SI.train))]),
                        xreg=cbind(cov1,cov2),
                        ic="aicc",trace=TRUE)
SI_arimax.predict = forecast(SI.arimax,h=10,xreg=cbind(c((SO.train[-(1:(length(SO.train)-So_ind))]),
                                                          (SO_best[,c(Code)][1:(10-So_ind)])),
                                                        c((Ext.train[-(1:(length(Ext.train)-Ext_ind))]),
                                                          (Extpredict$pred))))
  SI_arimax.pred = as.matrix(SI_arimax.predict$mean)

  summary(SI.arimax)

  SIResult[,7]=SI_arimax.pred[(length(SI_arimax.pred)-9):length(SI_arimax.pred)]

  err = rep(NA,8)
  for (j in (1:8))
  {
    err[j]=abs(SI_arimax.pred[j]-SI.test[j])/SI.test[j]
  }
  error[7]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3+err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7)/2

  error

Result = data.frame(matrix(c(0:0),nrow = 41,ncol = 31))
colnames(Result)<-colnames(Nob_SI_Mon_Dose)[2:32]
for(i in (1:31)){
  pred = c(Nob_SI_Mon_Dose[(1:31),(i+1)],rep(0,10))
  b = sum(Nob_SI_Mon_Dose$Aggregate)/sum(Nob_SI_Mon_Dose[,(i+1)])
  SO_.pred = SO_best$Aggregate/b
  Ext_.pred = c(Ext_.train,Extpredict$pred)/b
  for(j in (32:41)){
    pred[j] = 2.1734*(SO_.pred[j]) + 0.0078*(Ext_.pred[j-1])
  }
  Result[,i]=as.matrix(pred)
}
```

10 - Activyl Monthly Sales Out Pre-Processing

```
SO<-read.csv(file="Nobivac&Activyl_Sales_2013-2015.csv",header=TRUE)
colnames(SO)[colnames(SO)=="PRODUCT_CODE"]<-"Sku_Code"
colnames(SO)[colnames(SO)=="PRODUCT_NAME"]<-"Sku_Name"
colnames(SO)[colnames(SO)=="PRODUCT_FAMILY"]<-"GPF_Desc"
SO$Date<-as.Date(SO$INVOICE_DATE,"%d%b%Y")
SO$Month<-as.Date(cut(SO$Date,breaks="month"))
SO$Sku_Code<-as.factor(SO$Sku_Code)
a = grep("ACTIVYL",SO$Sku_Name)
A_SO <- SO[a,]
A_SO_Qty<-aggregate(A_SO$QUANTITY,by=list(A_SO$Month,A_SO$Sku_Code),sum)
names(A_SO_Qty)<-c("Month","Sku_Code","Quantity")
library(reshape)
A_SO_Qty<-cast(A_SO_Qty,Month~Sku_Code,sum,value="Quantity")[(1:18),]

for (j in (2:25)){
  for (i in (2:17)){
    if (A_SO_Qty[i,j]<=0){
      A_SO_Qty[i,j] = 0.5*(A_SO_Qty[(i-1),j]+A_SO_Qty[(i+1),j])
    }
  }
}
```

11 - Activyl Net Sales In and Net Sales Out from Inventory

```
inventory = read.csv("Inventory_Dist.csv",header = TRUE)
inventory$INV_DT = as.Date(inventory$INV_DT,"%m/%d/%Y")
```

```r
inventory$Year<-format(inventory$INV_DT,"%Y")
inventory$Month<-as.Date(cut(inventory$INV_DT,breaks="month"))
inventory$Day<-format(inventory$INV_DT,"%d")
# inventory<-inventory[which(inventory$Year!="2013"),]
inventory<-inventory[which(inventory$Month!="2015-12-01"),]

warehouse = unique(inventory$DIST_CODE)
setwd("C:/Users/angie/Desktop/EY-Merck/After Mid-term/Activyl")
A_convert<-read.csv(file="DosageConversion_Activyl.csv",header=TRUE)
skuCode = unique(A_convert$Sku_Code)
dateSequence = seq(as.Date("2013/01/01"),as.Date("2015/11/30"), by = "day")

z.Ind_P = zoo(rep(NA,length(dateSequence)),order.by=dateSequence)
z.Ind_S = zoo(rep(NA,length(dateSequence)),order.by=dateSequence)

z.Full_P = zoo(rep(NA,length(dateSequence)),order.by=dateSequence)
z.Full_S = zoo(rep(NA,length(dateSequence)),order.by=dateSequence)

Sku<-list()
for (k in (1:length(skuCode)))
{
 Sku[[k]]=inventory[grep(skuCode[k],inventory$MFGPARTNUMBER),]
 uniqueWarehouse=unique(Sku[[k]]$DIST_CODE)
 C = toString(skuCode[k])

 Sku_Dis<-list()
 z.P = zoo(rep(NA,length(dateSequence)),order.by=dateSequence)
 z.S = zoo(rep(NA,length(dateSequence)),order.by=dateSequence)

 for (j in (1:length(uniqueWarehouse))){
  Sku_Dis[[j]]=Sku[[k]][grep(uniqueWarehouse[j],Sku[[k]]$DIST_CODE),]
  W = toString(uniqueWarehouse[j])
  Sku_Dis[[j]]=Sku_Dis[[j]][order(Sku_Dis[[j]]$INV_DT),]
  n=length(Sku_Dis[[j]]$INV_DT)
  I <- rep(0,n); P <- rep(0,n); S <- rep(0,n)
  for (i in (1:n)){
    I[i]=Sku_Dis[[j]]$SUM_QUANTITY_[i]
  }
  for (i in (2:n)){
    if (I[i]<I[i-1]){
      S[i]=I[i-1]-I[i]
      P[i]=0
    }
   else{
      P[i]=I[i]-I[i-1]
      S[i]=0
    }
  }

  m.P<-as.matrix(P)
  colnames(m.P)<-paste(W)
  y.P<-zoo(m.P,order.by=Sku_Dis[[j]]$INV_DT)
  z.P<-merge(z.P,y.P)

  m.S<-as.matrix(S)
  colnames(m.S)<-paste(W)
  y.S <-zoo(m.S,order.by=Sku_Dis[[j]]$INV_DT)
  z.S<-merge(z.S,y.S)
  }

  p=zoo(rowSums(z.P,na.rm=TRUE),time(z.P))
  s=zoo(rowSums(z.S,na.rm=TRUE),time(z.S))

  CumP = cumsum(p)
  CumS = cumsum(s)
  m.Cum = as.matrix(merge(CumP,CumS))
  ratio = as.vector(m.Cum[,1]/m.Cum[,2])
  start_ind = min(which(ratio[-(1:365)]<1.1 & ratio[-(1:365)]>0.9))

  ind_P = p[-(1:365),][-(1:(start_ind-1)),]
  Core_ind_P = as.matrix(coredata(ind_P))
  colnames(Core_ind_P)<-paste(C)
  ind_P<-zoo(Core_ind_P,order.by=time(ind_P))
  z.Ind_P <- merge(z.Ind_P,ind_P)

  ind_S = s[-(1:365),][-(1:(start_ind-1)),]
  Core_ind_S = as.matrix(coredata(ind_S))
  colnames(Core_ind_S)<-paste(C)
  ind_S<-zoo(Core_ind_S,order.by=time(ind_S))
  z.Ind_S <- merge(z.Ind_S,ind_S)

  Core_full_P = as.matrix(coredata(p))
  colnames(Core_full_P)<-paste(C)
  full_P<-zoo(Core_full_P,order.by=time(p))
  z.Full_P<-merge(z.Full_P,full_P)

  Core_full_S = as.matrix(coredata(s))
  colnames(Core_full_S)<-paste(C)
  full_S<-zoo(Core_full_S,order.by=time(s))
  z.Full_S<-merge(z.Full_S,full_S)
}
```

12 - Activyl Sales-In Forecast SKU-Level

```
NetSO_full.z<-aggregate(z.Full_S,as.yearmon)
cum_NetSO.z<-cumsum(NetSO_full.z)
cum_NetSO<-as.data.frame(cum_NetSO.z)[,-1]
NetSI_full.z<-aggregate(z.Full_P,as.yearmon)
cum_NetSI.z<-cumsum(NetSI_full.z)
cum_NetSI<-as.data.frame(cum_NetSI.z)[,-1]

Actual_NetSI<-cum_NetSI
for (i in (1:24)){
  {
    for (j in (2:35)){
      Actual_NetSI[j,i]=cum_NetSI[j,i]-cum_NetSI[(j-1),i]
    }
  }
}

A_SI.mat<-data.matrix(Actual_NetSI)[(1:18),(1:24)]
library(zoo)
z.A_SI<-zoo(A_SI.mat,A_SI_Qty$Month[1:18])
library(lattice)
xyplot(z.A_SI,type="o",main="Monthly Sales In Qty for Activyl Skus from 2013-1 to 2014-06",
       strip = function(bg,...)
         strip.default(bg='white',...))

A_convert<-read.csv(file="DosageConversion_Activyl.csv",header=TRUE)
skuCode = unique(A_convert$Sku_Code)
```

```r
for (i in (1:length(skuCode))){
  mypath<-file.path("C:","Users","angie","Desktop",
                    "Inventory Plot (18 Months)",
                    paste("Cum P and S of ",skuCode[i],".jpg",sep=""))
  jpeg(file=mypath)
  mytitle=paste("Cumutative Net Sales In vs Cumulative Net Sales Out of ",skuCode[i],sep="")
  plot(as.vector(cum_NetSO[,i]),as.vector(cum_NetSI[,i]),
       xlab="Cumulative Net Sales Out", ylab="Cumulative Net Sales In",
       main=list(mytitle,cex=0.7))
  abline(1,1,col="red")
  leg.text<-c("Y=X")
  legend("topleft",inset=0.01,legend=leg.text,col=c("red"),lty=c(1),cex=0.8)
  dev.off()
}

Cum_ActualSO = cumsum(A_SOResult)

ActualSI =data.frame(matrix(c(0:0),nrow = 18,ncol = length(A_convert$Sku_Code)))
colnames(ActualSI)=skuCode
ErrorSI =data.frame(matrix(c(0:0),nrow = 1,ncol = length(A_convert$Sku_Code)))
colnames(ErrorSI)=skuCode

for (i in c(1,2,3,4,6,7,8,9,10,11,13,14,15,16,17,18,19,20,21,22,23)){
  i=18
  NSI = cum_NetSI[(1:7),i]
  NSO = cum_NetSO[(1:7),i]

  library(splines)
  fit=smooth.spline(NSO,NSI,df=3)
  plot(NSO,NSI)
  lines(fit,col="red")

  plot(as.vector(cum_NetSO[,i]),as.vector(cum_NetSI[,i]),
       xlab="Cumulative Net Sales Out",
       ylab="Cumulative Net Sales In")
  points(NSO,predict(fit)$y,type="l",col="blue")
  abline(1,1,col="red")

#   s = sum(Cum_ActualSO[1:6,i])/sum(NSO)
#   NSO = c(NSO,Cum_ActualSO[7:18,i]/s)
#   NSI_.pred = predict(fit,as.data.frame(NSO))
#   Cum_ActualSI = NSI_.pred$y*s

  s = sum(Cum_ActualSO[1:7,i])/sum(NSO)
  NSO = c(NSO,Cum_ActualSO[8:18,i]/s)
  NSI_.pred = predict(fit,as.data.frame(NSO))$y
  NSI_.pred = as.vector(NSI_.pred[,1])
  Cum_ActualSI = NSI_.pred*s

  par(mar=c(3,5,3,2))
  dev.off()
  plot(as.vector(cum_NetSO[,i]),
       as.vector(cum_NetSI[,i]),xlab="Cumulative Net Sales Out",
       ylab="Cumulative Net Sales In")
  points(NSO,NSI_.pred,type="l",col="green")
  abline(1,1,col="red")
  leg.text<-c("fitted line","Y=X")
  legend("topleft",inset=0.01,legend=leg.text,col=c("green","red"),lty=c(1),cex=0.4)

  ratio = as.vector(NSI_.pred/NSO)
  if(sum(ratio[-(1:6)]<1.1&ratio[-(1:6)]>0.9)==0) {
    Index = 18
  } else {
    start_ind = min(which(ratio[-(1:6)]<1.1 & ratio[-(1:6)]>0.9))
    Index = 6+start_ind}

  if(Index<=18){
    NSI_.pred = c(NSI_.pred[1:(Index-1)],NSO[Index:18])
    Cum_ActualSI = c(Cum_ActualSI[1:(Index-1)],Cum_ActualSO[Index:18,i])
  } else{
    NSI_.pred = NSI_.pred[1:Index]
    Cum_ActualSI = Cum_ActualSI[1:Index]
  }

   err = rep(NA,12)
  for (j in (1:12))
  {
    err[j]=abs(NSI_.pred[j+6]-cum_NetSI[(j+6),i])/cum_NetSI[(j+6),i]
  }
  ErrorSI[1,i]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3
                +err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7
                +err[9]/2^8+err[10]/2^9+err[11]/2^10+err[12]/2^11)/1.9995

  ActualSI[1,i]=Cum_ActualSI[1]
  for(j in (2:18)){
    ActualSI[j,i]=Cum_ActualSI[j]-Cum_ActualSI[j-1]
  }
}

## 114162, 114997, 115627
for (i in c(5,12,24)){
  i=5
  NSI = cum_NetSI[,i][1:7]
  NSO = cum_NetSO[,i][1:7]

  library(splines)
  #fit = lm(NSI~ns(NSO,df=5))
  fit=smooth.spline(NSO,NSI,df=5)
  plot(NSO,NSI)
  lines(fit,col="red")

  plot(as.vector(cum_NetSO[,i]),as.vector(cum_NetSI[,i]),
       xlab="Cumulative Net Sales Out", ylab="Cumulative Net Sales In")
  points(NSO,predict(fit)$y,type="l",col="blue")
  abline(1,1,col="red")

  s = sum(Cum_ActualSO[1:7,i])/sum(NSO)
  NSO = c(NSO,Cum_ActualSO[8:18,i]/s)
  NSI_.pred = predict(fit,as.data.frame(NSO))$y
  NSI_.pred = as.vector(NSI_.pred[,1])
  Cum_ActualSI = NSI_.pred*s

  plot(as.vector(cum_NetSO[,i]),
       as.vector(cum_NetSI[,i]),xlab="Cumulative Net Sales Out",
       ylab="Cumulative Net Sales In")
  points(NSO,NSI_.pred,type="l",col="green")
  abline(1,1,col="red")


  ratio = as.vector(NSI_.pred/NSO)
  if(sum(ratio[-(1:6)]<1.1&ratio[-(1:6)]>0.9)==0) {
    Index = 18
  } else {
    start_ind = min(which(ratio[-(1:6)]<1.1 & ratio[-(1:6)]>0.9))
    Index = 6+start_ind}

  if(Index<=18){
    NSI_.pred = c(NSI_.pred[1:(Index-1)],NSO[Index:18])
    Cum_ActualSI = c(Cum_ActualSI[1:(Index-1)],Cum_ActualSO[Index:18,i])
  } else{
    NSI_.pred = NSI_.pred[1:Index]
    Cum_ActualSI = Cum_ActualSI[1:Index]
  }

  err = rep(NA,12)
  for (j in (1:12))
  {
    err[j]=abs(NSI_.pred[j+6]-cum_NetSI[(j+6),i])/cum_NetSI[(j+6),i]
  }
  ErrorSI[1,i]=(err[1]+err[2]/2^1+err[3]/2^2+err[4]/2^3
              +err[5]/2^4+err[6]/2^5+err[7]/2^6+err[8]/2^7
              +err[9]/2^8+err[10]/2^9+err[11]/2^10+err[12]/2^11)/1.9995

  ActualSI[1,i]=Cum_ActualSI[1]
  for(j in (2:18)){
    ActualSI[j,i]=Cum_ActualSI[j]-Cum_ActualSI[j-1]
  }
}

SIforecast.mat<-data.matrix(ActualSI)
library(zoo)
z.A_SI<-zoo(SIforecast.mat,A_SI_Qty$Month[1:18])
library(lattice)
xyplot(z.A_SI,type="o",main="Monthly Sales In Forecast for Activyl Skus (In QTY)",
       strip = function(bg,...)
         strip.default(bg='white',...))
```